(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 January 2004 (29.01.2004)

PCT

(10) International Publication Number
WO 2004/010346 A2

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/EP2003/007699

(22) International Filing Date: 16 July 2003 (16.07.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/199,544 19 July 2002 (19.07.2002) US

(71) Applicant: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant (*for LU only*): IBM DEUTSCHLAND GMBH [DE/DE]; Pascalstrasse 100, 70569 Stuttgart (DE).

(72) Inventors: BENKE, Oliver; Hauptstr. 44, 70771 Leinfelden-Echterdingen (DE). BETZLER, Boas; Seestr. 21-1, 71106 Magstadt (DE). LUMPP, Dr. Thomas; Melanchthonstr. 21, 72770 Reutlingen (DE). PASCH, Eberhard; Elly-Heuss-Knapp-Str. 21, 72074 Tuebingen (DE).

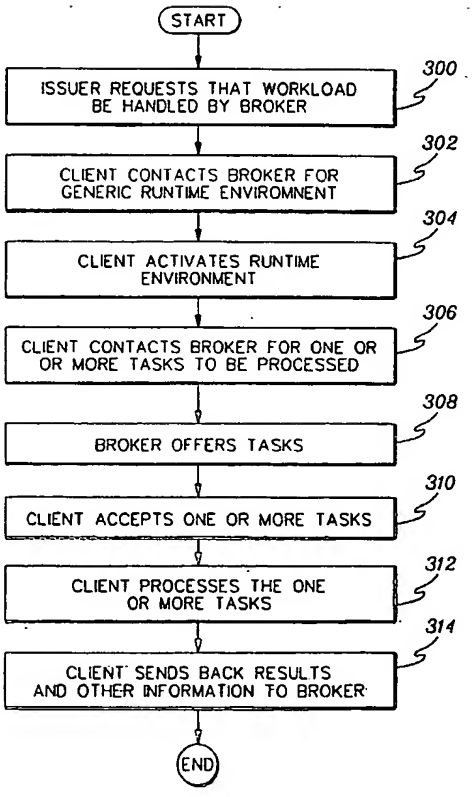
(74) Agent: DUSCHER, Reinhard; IBM Deutschland GmbH, Intellectual Property, 70548 Stuttgart (DE).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: EMPLOYING A RESOURCE BROKER IN MANAGING WORKLOADS OF A PEER-TO-PEER COMPUTING ENVIRONMENT



(57) Abstract: A resource broker is used in managing workloads of a peer-to-peer distributed computing environment. The broker is responsible for distributing to a plurality of clients of the peer-to-peer distributed computing environment tasks of one or more workloads of one or more issuers of the environment. The tasks are processed by the clients using a runtime environment provided by the broker. The runtime environment of a client is generic to the one or more issuers and independent of an operating system of the client.

WO 2004/010346 A2

Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

DESCRIPTION

EMPLOYING A RESOURCE BROKER IN MANAGING WORKLOADS OF A PEER-TO-PEER COMPUTING ENVIRONMENT

Technical Field

This invention relates, in general, to a peer-to- peer distributed computing environment, and in particular, to employing a resource broker in managing workloads of the peer-to-peer distributed computing environment.

Background of the Invention

In one embodiment of a peer-to-peer distributed computing environment, one or more issuers have a direct relationship with one or more client resource providers. In such an environment, an issuer, desiring to have a workload processed, partitions the workload into a plurality of tasks, and delegates the tasks to the one or more client resource providers. The client resource providers accept the tasks and process them using local resources. The providers then send back the results of the tasks to the issuer.

In this environment, in order for a client resource provider to process a task, the client needs to contact the issuer to download and install a runtime environment to be used in processing the task. The runtime environment is specific to the issuer, and thus, if the issuer changes, so does the runtime environment. Further, if the client resource provider wishes to support multiple issuers, then multiple runtime environments need to be downloaded and installed on the client.

Based on the foregoing, a need exists for a peer- to-peer distributed computing environment that facilitates the processing of workloads by clients. For example, a need exists for an environment in which changes of the issuer do not affect the runtime environment. Further, a need exists for an environment in which a direct relationship is not necessary between the issuers and clients. Yet further, a need exists for an environment that supports multiple issuers without downloading and installing multiple runtime environments.

Summary of the Invention

The shortcomings of the prior art are overcome and additional advantages are provided through the provision of a method of managing workloads of a peer-to-peer distributed computing environment. The method includes, for instance, distributing, by a broker of the peer-to-peer distributed computing environment to a plurality of clients of the peer- to-peer distributed computing environment, a plurality of tasks of a workload of an issuer of the peer-to-peer distributed computing environment; and processing the plurality of tasks by the plurality of clients, wherein a client of the plurality of clients uses a runtime environment for the processing, the runtime environment being generic to the issuer and independent of an operating system of the client.

Further, a method of facilitating management of workloads of a peer-to-peer distributed computing environment is provided. The method includes, for instance, managing via a resource broker one or more workloads of one or more issuers of the peer-to-peer distributed computing environment, the peer-to-

peer distributed computing environment comprising a plurality of clients to process at least one workload of the one or more workloads.

Yet further, a method of facilitating processing of workloads of a peer-to-peer distributed computing environment is provided. The method includes, for instance, obtaining by a client of a plurality of clients of a peer- to-peer distributed computing environment one or more tasks of a workload of an issuer of the peer-to-peer distributed computing environment; and processing by the client the one or more tasks, wherein the client uses a runtime environment in processing, the runtime environment being generic to the issuer and independent of an operating system of the client.

System and computer program products corresponding to the above-summarized methods are also described herein.

Additional features and advantages are realized through the techniques of the present invention. Other embodiments and aspects of the invention are described in detail herein and are considered a part of the claimed invention.

Brief Description of the Drawings

The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

FIG. 1 depicts one example of a peer-to- peer distributed computing environment incorporating and using one or more aspects of the present invention;

FIG. 2 depicts one example of further details of a client resource provider of FIG. 1, in accordance with an aspect of the present invention;

FIGs. 3a-3b depict one embodiment of the logic associated with using a resource broker to process workloads of one or more issuers of a peer-to-peer distributed computing environment, in accordance with an aspect of the present invention; and

FIG. 4 depicts one embodiment of the logic associated with a runtime flow and a task flow, in accordance with an aspect of the present invention.

Best Mode for Carrying Out the Invention

In accordance with an aspect of the present invention, a resource broker (e.g., a grid computing resource broker) is used in managing workloads of a peer-to- peer distributed computing environment. The broker is provided as a layer between issuers and clients of the environment, and is responsible for receiving workloads from the issuers and having those workloads processed by the clients. The use of the resource broker enables the clients to process the workloads using a common or generic runtime environment.

One embodiment of a peer-to-peer distributed computing environment incorporating and using one or more aspects of the present invention is described with reference to FIG. 1. In one example, a peer-to-peer distributed computing environment

100 includes a resource broker 102 coupled to one or more issuers 104 and to one or more client resource providers 106 via, for instance, Internet Protocol (IP) network communications (e.g., HTTP). Resource broker 102 enables the one or more issuers to be coupled to the one or more client resource providers via an indirect relationship. That is, no direct relationship needs to exist between an issuer and a client.

In one example, the resource broker provides the infrastructure to receive workloads from one or more issuers, to distribute the workloads to one or more client providers, to track the workloads, and to perform accounting associated with the workloads, etc. The broker executes on a communications unit, such as, for example, a Power PC, a UNIX based mainframe, or a mainframe, to name just a few examples.

An issuer is, for instance, an application executing on a communications unit, such as, for example, an Intel or RISC based server. The application is typically, but not necessarily, computational intensive, and thus, desirous of executing within a peer-to-peer distributed environment. As examples, an issuer includes SETI, which is a search for extraterrestrial intelligence; or Folding@home, a protein folding application for drug research; or numerous other applications.

The issuer partitions its workload into one or more tasks to be processed. As examples, for business applications, a workload can be partitioned on a transaction basis, and for scientific applications, a workload can be partitioned by splitting up the computing range. In one embodiment, a task is a unit of work that uses one or more resources (e.g., CPU,

main memory, non-volatile memory, I/O bandwidth) during processing. Tasks from one issuer can be normalized, such that they use the same amount of resources, or they can be undetermined, such that they use differing amounts of resources. The tasks are processed by one or more client resource providers.

A client resource provider is, for instance, a communications unit, such as, for example, a game station, a cell phone, an Intel personal computer, a server (e.g., an RS/6000), a mainframe or a Personal Digital Assistant (PDA). The providers use their own local resources to process the tasks, and are capable of processing tasks of one or more workloads of one or more issuers.

In one example, each client is isolated from one another and from the issuers. Further, each issuer is isolated from one another. Additionally, the clients may be homogeneous or at least two of the clients of the distributed environment may be heterogeneous to one another. For instance, at least two clients may have different architectures, different operating systems, and/or are different kinds of units (e.g., PDA, cell phone, personal computer, server, mainframe, etc.).

Further details regarding a client resource provider are described with reference to FIG. 2. In one example, a client resource provider 106 includes a central processing unit 200 executing an operating system, a disk storage 202, and one or more client applications 204. Additionally, in accordance with an aspect of the present invention, client 106 includes a common (or generic) runtime environment 206 used in processing one or more tasks 208 of one or more workloads. The common

runtime environment co- exists with and is independent of the general purpose operating system of the CPU.

As described in further detail below, in one example, common runtime environment 206 is a constrained and isolated environment, which is defined by a user of the client. The user configures the environment by indicating which and how many resources of the client can be acquired by the common runtime environment and by the one or more tasks executing in the common runtime environment. The resources set aside for the common runtime environment are isolated from other applications and resources of the client.

The common runtime environment is a generic environment usable in processing tasks for multiple and diverse issuers. The runtime environment is, for example, a Java Virtual Machine or an operating system of a virtual machine that is generic to the issuers. The runtime environment is independent of the issuers and the clients. If an issuer changes, there is no effect on the runtime environment. Thus, the generic runtime environment need not be reinstalled for different issuers. Further, the runtime environment need not be reinstalled each time a task or a different workload is to be performed.

In one embodiment, the common runtime environment is provided to the clients by the resource broker. This processing, as well as other processing associated with the resource broker and a client provider, are described with reference to FIGs. 3a, 3b and 4.

One embodiment of the logic associated with utilizing a resource broker to manage the processing of workloads of a peer-to-peer distributed computing environment is described

the client the best rate of return. For example, if a particular client is CPU intensive, but has less memory, then the client looks to those tasks that need more CPU but less memory. Further, the client decides which of those tasks has a better rate of return.

Subsequent to making the analysis, the client accepts one or more tasks for processing, STEP 310. In one embodiment, the tasks and any associated data objects are encrypted (e.g., using Secure Socket Layer (SSL)) by the broker and then forwarded to the client. The client runtime environment decrypts the tasks and data, and processes the one or more tasks using, for instance, local resources of the client, STEP 312. Additional details of this processing is described further below with reference to FIG. 4.

During processing, the runtime environment tracks the amount of resources consumed by the client to perform the one or more tasks. In one example, it tracks the particular resources consumed by each task.

Upon completion of processing the one or more tasks, the client sends back to the resource broker the results of the processing, as well as information about resource consumption, STEP 314. Based on this information, the broker pays the client for each completed task, STEP 316 (FIG. 3b). The payment is based, for example, on the resources consumed, as well as on the agreed rate.

The broker accumulates the results for a workload from one or more client providers, STEP 318. The broker is responsible for tracking which tasks have been distributed to which client and which tasks have been completed. When the broker receives

the results for all of the tasks of a workload, the broker determines one or more end results. Eventually, the broker provides the one or more end results to the issuer, and the issuer pays the broker, STEP 320.

In one embodiment, the broker keeps track of which clients, from a historic perspective, have worked on which tasks and for how long. At any given time, the broker knows how much of the workload has been processed. It also has, in one embodiment, a timeout mechanism, such that a task distributed to the client timeouts if it has not been processed or has been processing for a selected period of time.

With the above logic, the broker can service one or more issuers at any one time. Further, one or more clients can receive and activate the runtime environment and process tasks using that runtime environment. In one example, a client provider continues to request tasks from the broker and process those tasks, until the client deactivates the runtime environment.

Further details associated with the use of the runtime environment in processing one or more tasks is described with reference to FIG. 4. In particular, FIG. 4 depicts two logic flows: one occurring in the runtime environment, and another associated with a task being processed. In one example, the common runtime environment has the ability to control the amount of resources used for a particular task, based on user input of the client activating the runtime environment. To demonstrate this ability, an example is described herein in which memory is being controlled. The common runtime environment implements its own memory management subsystem for

the task and caps the amount of memory resource utilized by the task.

Initially, the runtime environment is started on the client, STEP 400, and during an initialization phase, the runtime environment allocates a predefined amount of memory of the client usable by the runtime environment, STEP 402. Thus, the runtime environment is limited to using the allocated amount of memory. No additional memory is allocated to the runtime environment. Further, this allocated memory is not available to any other applications on the client, thus, providing isolation, privacy and security of the data in the memory.

Subsequent to allocating the memory, the memory is cleared (e.g., overwritten with zeros), STEP 404. This provides isolation for one or more tasks that are to use the allocated memory. Additionally, a task is started, STEP 406, and the task allocates one or more objects in memory, STEP 408. In one example, in order to accomplish the allocation, the task contacts the runtime environment, which provides the allocation function for the task. Since the memory accessed by the task is controlled by the runtime environment, it is guaranteed to have isolation and protection. (In a further embodiment, if a data space is not needed, then it is not allocated, but the computation is performed (see reference numeral 409).)

The runtime environment attempts to reserve the desired amount of the allocated memory pool, STEP 410. If an insufficient amount of memory is available, INQUIRY 412, then an error condition is encountered, STEP 413. In response to encountering the error condition, various steps can be taken.

For instance, the task can terminate, and/or an error handling condition can be employed.

However, if memory is available in the pool, then the reserved amount of memory (i.e., data space) is usable by the task, STEP 414. The task performs one or more functions (e.g., computations) using the data space. After use of the data space is complete, the data space is freed,

STEP 416. In one embodiment, this includes unreserving the memory by the runtime environment, STEP 418. Further, the task ends, STEP 420.

Any outstanding memory is then un-reserved, STEP 424, and the memory is cleared, STEP 426. Additionally, the memory is deallocated, STEP 426.

In the aboye example, resource capping is achieved by the fact that the runtime environment only allocates as much memory as configured by the end user. Thus, the task thereafter can only allocate memory out of the pool that was originally acquired by the runtime environment. Isolation, in this example, is achieved by the fact that the memory is cleared before and after the tasks use the memory. Also, during the duration of the task, memory is owned by the runtime environment, and the client does not allow other applications running on the client to access the memory in use by the common runtime environment. Since the common runtime environment implements its own memory management, the tasks cannot access memory other than that allocated by the common runtime environment.

In other embodiments, other types of resources can be capped. For example, CPU utilization can also be capped by again indicating at the common runtime environment the amount of CPU utilization to be used by each task. Other resources, such as I/O bandwidth, can also be constrained. As one example, further details regarding capping are provided through a product offered by International Business Machines Corporation referred to as the Intelligent Resource Director on z/OS and references associated therewith.

Described in detail above is the use of a resource broker in managing the processing of workloads in a peer-to- peer distributed computing environment. The resource broker provides a common runtime environment to the clients that are to process tasks of the workloads. Advantageously, issuers do not need to build a relationship with the clients. The issuers obtain results faster by working with the established broker, instead of building one or more client communities. Client resources can be deployed for multiple issuers without changing the runtime environment, and the client runtime environment determines the best rate of return for issuer tasks. Integrity, security and privacy of tasks are protected.

Advantageously, one or more aspects of the present invention standardize the runtime environment, which provides flexibility (e.g., multiple tasks without reinstall), privacy and integrity of tasks, and security (e.g., client is secure from malicious task attacks). It also provides common metrics between issuers that can be used for standardized charging of resource consumption.

The embodiments described above are only examples. For instance, although examples of communications units are provided for issuers, clients and the broker, one or more aspects of the invention are not limited to such examples. Other communications units, as well as other units, are also possible. Further, one or more communications units can be included on one or more nodes of the environment.

The present invention can be included in an article of manufacture (e.g., one or more computer program products) having, for instance, computer usable media. The media has

embodied therein, for instance, computer readable program code means for providing and facilitating the capabilities of the present invention. The article of manufacture can be included as a part of a computer system or sold separately.

Additionally, at least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform the capabilities of the present invention can be provided.

The flow diagrams depicted herein are just examples. There may be many variations to these diagrams or the steps (or operations) described therein without departing from the spirit of the invention. For instance, the steps may be performed in a differing order, or steps may be added, deleted or modified. All of these variations are considered a part of the claimed invention.

Although preferred embodiments have been depicted and described in detail herein, it will be apparent to those skilled in the relevant art that various modifications, additions, substitutions and the like can be made without departing from the spirit of the invention and these are therefore considered to be within the scope of the invention as defined in the following claims.

C L A I M S

1. A method of managing workloads of a peer-to-peer distributed computing environment, said method comprising:

distributing, by a broker of the peer-to-peer distributed computing environment to a plurality of clients of the peer-to-peer distributed computing environment, a plurality of tasks of a workload of an issuer of the peer-to-peer distributed computing environment; and

processing the plurality of tasks by the plurality of clients, wherein a client of the plurality of clients uses a runtime environment for the processing, said runtime environment being generic to the issuer and independent of an operating system of the client.

2. The method of claim 1, further comprising providing by the issuer to the broker the workload to be distributed.

3. The method of claim 1, wherein the issuer is isolated from the plurality of clients.

4. The method of claim 1, wherein the runtime environment of the client provides isolation from other portions of the client.

5. The method of claim 1, wherein the runtime environment of the client facilitates in capping use of one or more

resources used in processing one or more tasks of the plurality of tasks.

6. The method of claim 1, wherein the issuer is coupled to the plurality of clients via the broker, thereby providing an indirect relationship between the issuer and the plurality of clients.

7. The method of claim 1, further comprising forwarding by one or more clients of the plurality of clients results of the processing to the broker.

8. The method of claim 7, further comprising:

accumulating the results by the broker; and

providing to the issuer one or more end results based on the accumulating.

9. The method of claim 1, further comprising providing by the broker payment to at least one client of the plurality of clients for the processing.

10. The method of claim 1, further comprising tracking by the broker which tasks of the plurality of tasks are being processed by which clients of the plurality of clients.

11. The method of claim 1, further comprising determining by a client of the plurality of clients one or more tasks of the plurality of tasks to be processed by that client.

12. The method of claim 11, wherein said determining is based on at least one of a profile of the client and a rate of return for the client.

13. The method of claim 1, further comprising providing payment to the broker from the issuer.

14. A method of facilitating management of workloads of a peer-to-peer distributed computing environment, said method comprising:

managing via a resource broker one or more workloads of one or more issuers of the peer-to- peer distributed computing environment, said peer- to-peer distributed computing environment comprising a plurality of clients to process at least one workload of the one or more workloads.

15. The method of claim 14, wherein one or more clients of the plurality of clients include a runtime environment, said runtime environment being generic to the one or more issuers.

16. The method of claim 14, wherein the managing comprises tracking a workload of the one or more workloads.

17. The method of claim 14, wherein the managing comprises providing one or more accounting services relating to a workload of the one or more workloads.

18. The method of claim 14, wherein the managing comprises distributing one or more tasks of the at least one

workload to one or more clients of the plurality of clients.

19. The method of claim 18, wherein the managing further comprises providing to the one or more clients a runtime environment to be used in the processing, said runtime environment being generic to the one or more issuers.

20. A method of facilitating processing of workloads of a peer-to-peer distributed computing environment, said method comprising:

obtaining by a client of a plurality of clients of a peer-to-peer distributed computing environment one or more tasks of a workload of an issuer of the peer-to-peer distributed computing environment; and

processing by the client the one or more tasks, wherein the client uses a runtime environment in processing, said runtime environment being generic to the issuer and independent of an operating system of the client.

21. A system of managing workloads of a peer-to-peer distributed computing environment, said system comprising:

a broker of the peer-to-peer distributed computing environment to distribute to a plurality of clients of the peer-to-peer distributed computing environment a plurality of tasks of a workload of an issuer of the peer-to-peer distributed computing environment; and

the plurality of clients to process the plurality of

tasks, wherein a client of the plurality of clients uses a runtime environment for the processing, said runtime environment being generic to the issuer and independent of an operating system of the client.

22. The system of claim 21, wherein the runtime environment of the client provides isolation from other portions of the client.

23. The system of claim 21, wherein the runtime environment of the client facilitates in capping use of one or more resources used in processing one or more tasks of the plurality of tasks.

24. The system of claim 21, wherein the issuer is coupled to the plurality of clients via the broker, thereby providing an indirect relationship between the issuer and the plurality of clients.

25. The system of claim 21, wherein one or more clients of the plurality of clients are adapted to forward results of the processing to the broker.

26. The system of claim 25, wherein the broker is adapted to accumulate the results and provide to the issuer one or more end results based on the accumulating.

27. The system of claim 21, wherein the broker is further adapted to track which tasks of the plurality of tasks are being processed by which clients of the plurality of clients.

28. A system of facilitating management of workloads of a peer-to-peer distributed computing environment, said system comprising:

a resource broker to manage one or more workloads of one or more issuers of the peer-to- peer distributed computing environment, said peer- to-peer distributed computing environment comprising a plurality of clients to process at least one workload of the one or more workloads.

29. The system of claim 28, wherein one or more clients of the plurality of clients include a runtime environment, said runtime environment being generic to the one or more issuers.

30. At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing workloads of a peer-to-peer distributed computing environment, said method comprising:

distributing, by a broker of the peer-to-peer distributed computing environment to a plurality of clients of the peer-to-peer distributed computing environment, a plurality of tasks of a workload of an issuer of the peer-to-peer distributed computing environment; and

processing the plurality of tasks by the plurality of clients, wherein a client of the plurality of clients uses a runtime environment for the processing, said runtime environment being generic to the issuer and independent of an operating system of the client.

31. The at least one program storage device of claim 30, wherein the runtime environment of the client provides isolation from other portions of the client.

32. The at least one program storage device of claim 30, wherein the runtime environment of the client facilitates in capping use of one or more resources used in processing one or more tasks of the plurality of tasks.

33. The at least one program storage device of claim 30, wherein the issuer is coupled to the plurality of clients via the broker, thereby providing an indirect relationship between the issuer and the plurality of clients.

34. The at least one program storage device of claim 30, wherein said method further comprises forwarding by one or more clients of the plurality of clients results of the processing to the broker.

35. The at least one program storage device of claim 34, wherein said method further comprises:

accumulating the results by the broker; and

providing to the issuer one or more end results based on the accumulating.

36. The at least one program storage device of claim 30, wherein said method further comprises tracking by the broker which tasks of the plurality of tasks are being processed by which clients of the plurality of clients.

37. At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of facilitating management of workloads of a peer-to-peer distributed computing environment, said method comprising:

managing via a resource broker one or more workloads of one or more issuers of the peer-to- peer distributed computing environment, said peer- to-peer distributed computing environment comprising a plurality of clients to process at least one workload of the one or more workloads.

38. The at least one program storage device of claim 37, wherein one or more clients of the plurality of clients include a runtime environment, said runtime environment being generic to the one or more issuers.

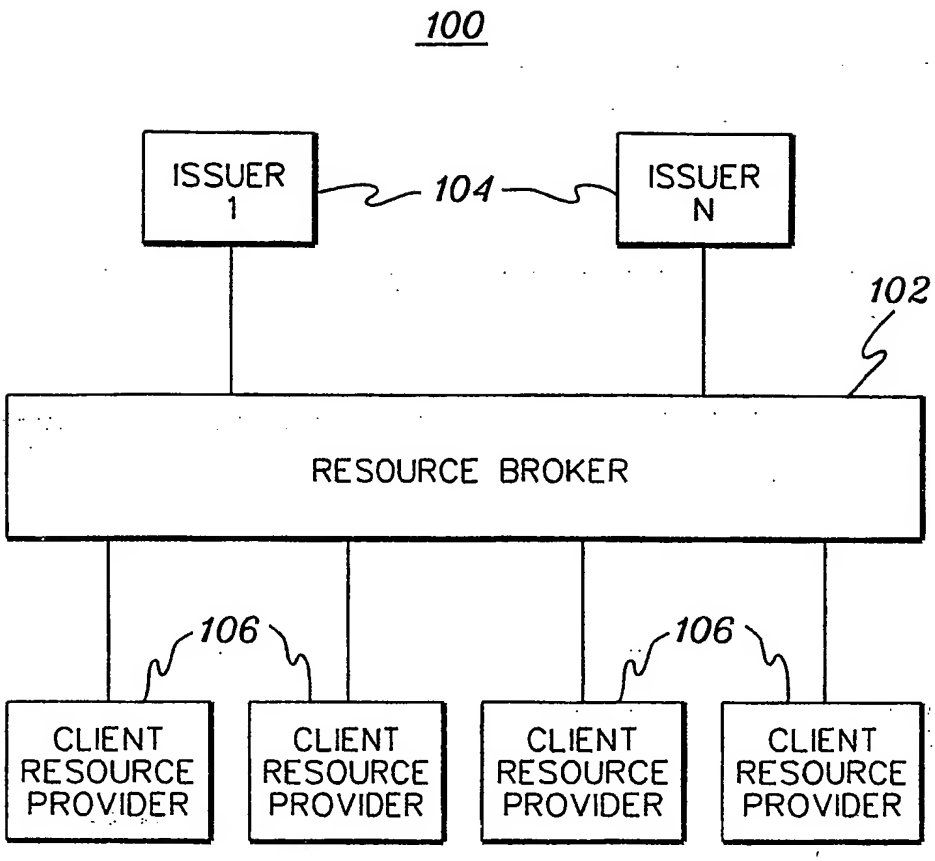

100
ISSUER 1
104
ISSUER N
102
RESOURCE BROKER
106
106
CLIENT RESOURCE PROVIDER
CLIENT RESOURCE PROVIDER
CLIENT RESOURCE PROVIDER
CLIENT RESOURCE PROVIDER

fig. 1

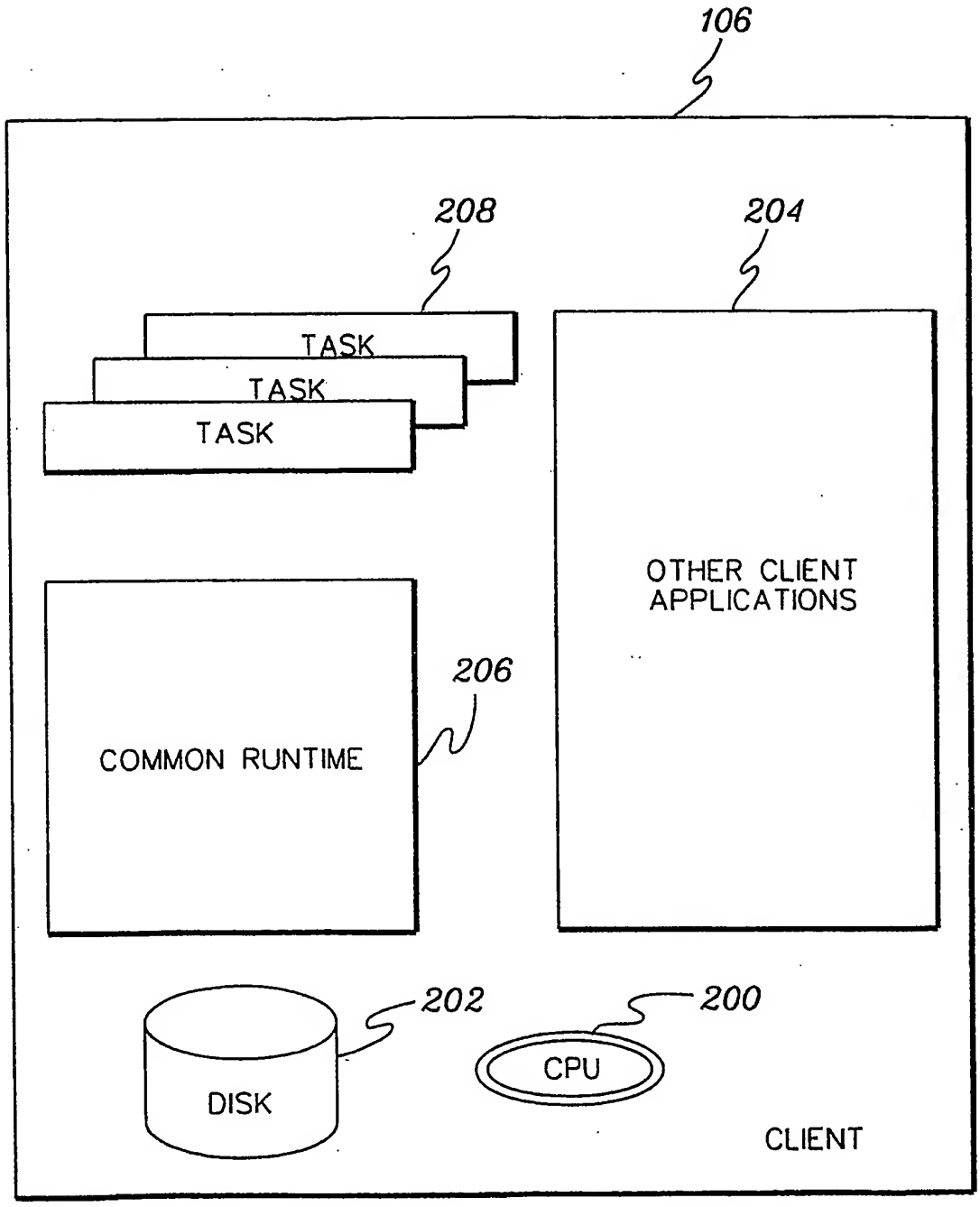

106
208
204
TASK
TASK
TASK
OTHER CLIENT APPLICATIONS
206
COMMON RUNTIME
202
200
DISK
CPU
CLIENT

fig. 2

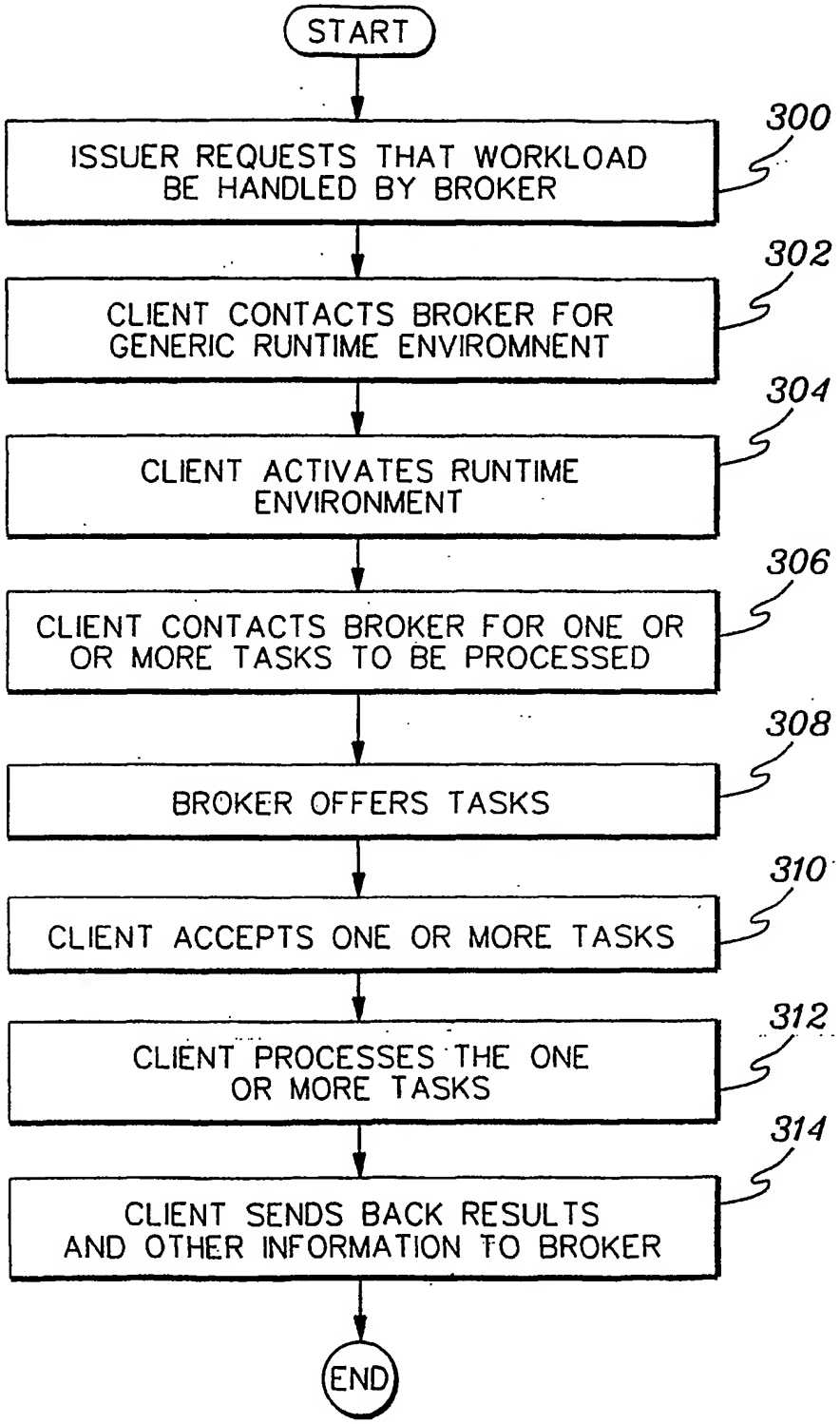

START
ISSUER REQUESTS THAT WORKLOAD BE HANDLED BY BROKER
300
CLIENT CONTACTS BROKER FOR GENERIC RUNTIME ENVIROMNENT
302
CLIENT ACTIVATES RUNTIME ENVIRONMENT
304
CLIENT CONTACTS BROKER FOR ONE OR OR MORE TASKS TO BE PROCESSED
306
BROKER OFFERS TASKS
308
CLIENT ACCEPTS ONE OR MORE TASKS
310
CLIENT PROCESSES THE ONE OR MORE TASKS
312
CLIENT SENDS BACK RESULTS AND OTHER INFORMATION TO BROKER
314
END

fig. 3A

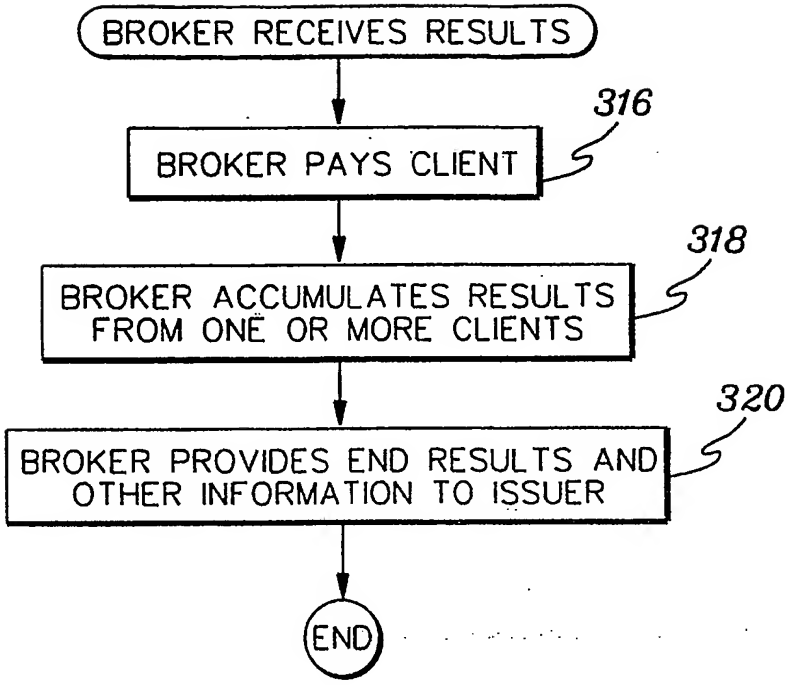

BROKER RECEIVES RESULTS
316
BROKER PAYS CLIENT
318
BROKER ACCUMULATES RESULTS FROM ONE OR MORE CLIENTS
320
BROKER PROVIDES END RESULTS AND OTHER INFORMATION TO ISSUER
END

fig. 3B

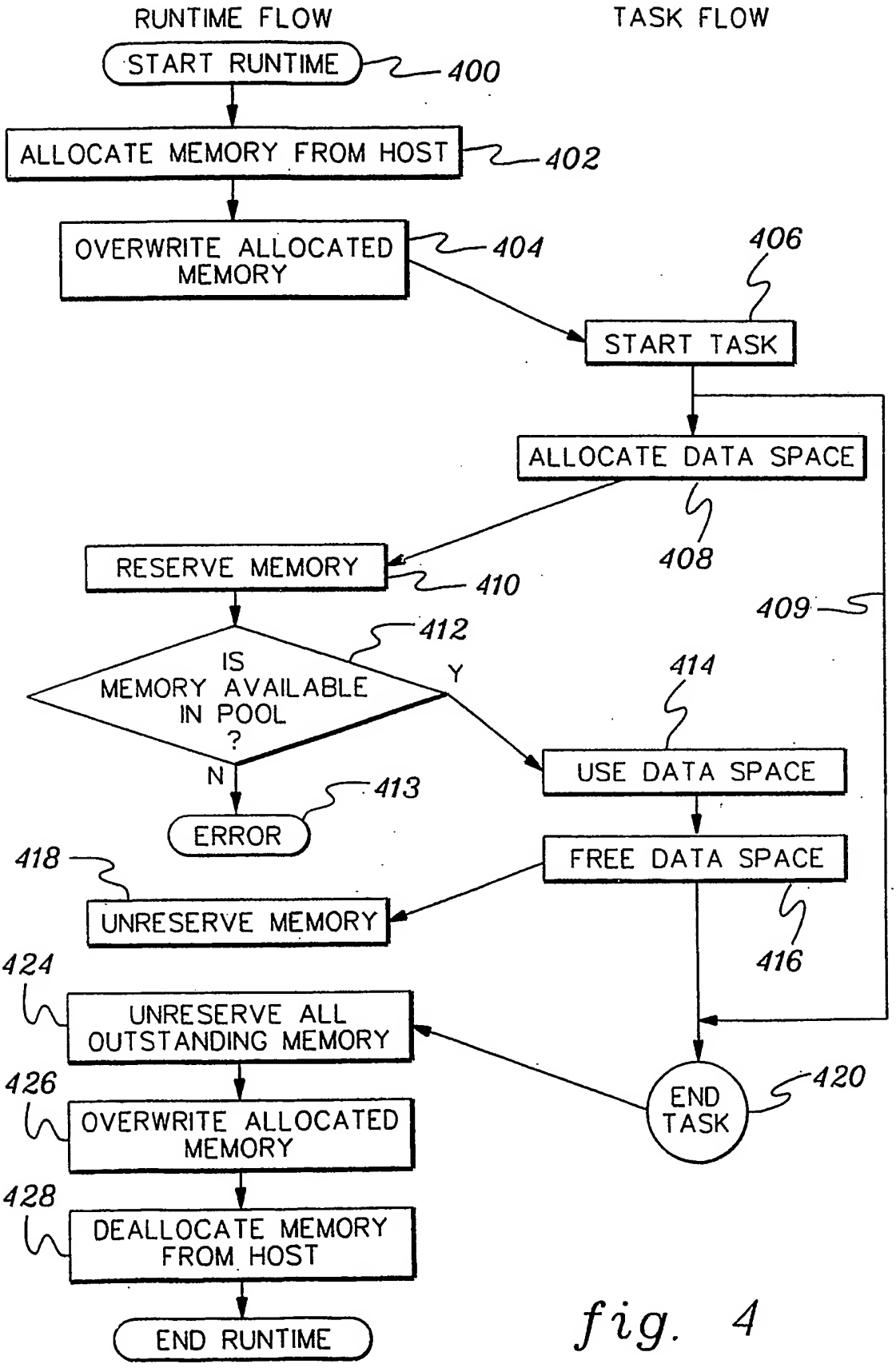

RUNTIME FLOW
TASK FLOW
START RUNTIME
400
ALLOCATE MEMORY FROM HOST
402
OVERWRITE ALLOCATED MEMORY
404
406
START TASK
ALLOCATE DATA SPACE
408
RESERVE MEMORY
410
409
412
IS MEMORY AVAILABLE IN POOL ?
Y
N
414
USE DATA SPACE
413
ERROR
FREE DATA SPACE
418
UNRESERVE MEMORY
416
424
UNRESERVE ALL OUTSTANDING MEMORY
END TASK
420
426
OVERWRITE ALLOCATED MEMORY
428
DEALLOCATE MEMORY FROM HOST
END RUNTIME

fig. 4

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 January 2004 (29.01.2004)

PCT

(10) International Publication Number
WO 2004/010346 A3

(51) International Patent Classification[7]: G06F 9/46

(21) International Application Number:
PCT/EP2003/007699

(22) International Filing Date: 16 July 2003 (16.07.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/199,544 19 July 2002 (19.07.2002) US

(71) Applicant: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant (*for LU only*): IBM DEUTSCHLAND GMBH [DE/DE]; Pascalstrasse 100, 70569 Stuttgart (DE).

(72) Inventors: BENKE, Oliver; Hauptstr. 44, 70771 Leinfelden-Echterdingen (DE). BETZLER, Boas; Seestr. 21-1, 71106 Magstadt (DE). LUMPP, Thomas; Melanchthonstr. 21, 72770 Reutlingen (DE). PASCH, Eberhard; Elly-Heuss-Knapp-Str. 21, 72074 Tuebingen (DE).

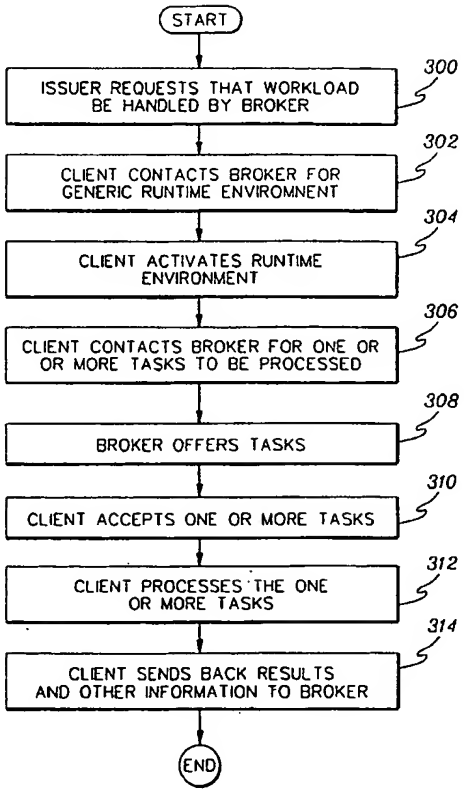
(74) Agent: DUSCHER, Reinhard; IBM Deutschland GmbH, Intellectual Property, 70548 Stuttgart (DE).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

[Continued on next page]

(54) Title: EMPLOYING A RESOURCE BROKER IN MANAGING WORKLOADS OF A PEER-TO-PEER COMPUTING ENVIRONMENT



(57) Abstract: A resource broker is used in managing workloads of a peer-to-peer distributed computing environment. The broker is responsible for distributing to a plurality of clients of the peer-to-peer distributed computing environment tasks of one or more workloads of one or more issuers of the environment. The tasks are processed by the clients using a runtime environment provided by the broker. The runtime environment of a client is generic to the one or more issuers and independent of an operating system of the client.

WO 2004/010346 A3

Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(88) Date of publication of the international search report:
9 December 2004

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 03/07699

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

'A' document defining the general state of the art which is not considered to be of particular relevance

'E' earlier document but published on or after the international filing date

'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

'O' document referring to an oral disclosure, use, exhibition or other means

'P' document published prior to the international filing date but later than the priority date claimed

'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

'&' document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 September 2004	27/10/2004

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	de Junca, I

Form PCT/ISA/210 (second sheet) (January 2004)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 01/14961 A (PARABON COMPUTATION) 1 March 2001 (2001-03-01) abstract; figures 1,3 page 3, lines 7-29	1,14,20, 21,28, 30,37
A	page 4, lines 8-14	11,12
X	page 7, lines 1-15; figure 1	2,3,6-8, 24-26, 33-35
X	page 8, lines 11-13,22-26; figure 2	10,11, 16,27,36
X	page 12, lines 1-12; figure 3	4,15,19, 22,29, 31,38
X	page 14, line 8 - page 15, line 28; figures 4A,4B	5,23,32
X	page 16, lines 22-27	7-9,13, 17,18, 25,26, 34,35

X	NEARY M O ET AL: "Javelin: Parallel computing on the internet" October 1999 (1999-10), FUTURE GENERATIONS COMPUTER SYSTEMS, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, NL, PAGE(S) 659-674 , XP004176754 ISSN: 0167-739X page 662, left-hand column, lines 22-32; figure 1 page 663, right-hand column, lines 14-29; figure 2	1-3,6,7, 14,15, 18-21, 24,25, 28-30, 33,34, 37,38
A	page 660, right-hand column, lines 19-38	4,5,22, 23,31,32
A	page 661, right-hand column, lines 37-43	12

X	US 6 112 225 A (LU QI ET AL) 29 August 2000 (2000-08-29) column 4, lines 7-12,31-37; figure 1 column 4, lines 49-53,62-67; figure 2 column 5, lines 3-8,38-44; figure 2 column 6, line 67 - column 7, line 42; figure 5	1-3,6, 14, 18-21, 24, 28-30, 33,37,38
X	column 10, lines 6-21	4,5,22, 23,31,32
X	column 4, line 67 - column 5, line 8	7,8,25, 26,34,35
X	Figure 5 (512) (514) column 7, lines 48-65	7,8,25, 26,34,35
A	column 9, lines 18-35; figure 6	11,12

	-/--	

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6 098 091 A (KISOR GREG) 1 August 2000 (2000-08-01) abstract; claims 1,12,13,22,24 column 3, lines 19-21,45-52; figure 1 column 7, lines 11-43; figure 5	1-38

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP 03/07699

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0114961 A	01-03-2001	AU 7114200 A	19-03-2001
		CA 2352807 A1	01-03-2001
		EP 1145109 A2	17-10-2001
		JP 2003507812 T	25-02-2003
		WO 0114961 A2	01-03-2001
		US 6463457 B1	08-10-2002
US 6112225 A	29-08-2000	NONE	
US 6098091 A	01-08-2000	NONE	